UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/650,894	08/28/2003	Jason Robert McGee	RSW920030102US1	8718

46320   7590   07/30/2009
CAREY, RODRIGUEZ, GREENBERG & PAUL, LLP
STEVEN M. GREENBERG
950 PENINSULA CORPORATE CIRCLE
SUITE 3020
BOCA RATON, FL 33487

EXAMINER
BATES, KEVIN T

ART UNIT	PAPER NUMBER
2456	

MAIL DATE	DELIVERY MODE
07/30/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 10/650,894
Filing Date: August 28, 2003
Appellant(s): MCGEE ET AL.

<div>

Steven M. Greenberg
Reg. No. 44,725
For Appellant

</div>

EXAMINER'S ANSWER

This is in response to the appeal brief filed June 23, 2009 appealing from the Office

action mailed January 23, 2009.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6519594	Li	2-2003
2004/0255293	Spotswood	12-2004
6721777	Sharma	4-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-2, 4, 6-12, and 14-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Li (6519594) in view of Spotswood (2004/0255293).

Regarding claims 1, 11, and 19, Li teaches a method of generating a logically merged web module for a web application, comprising:

Responsive to a determination that a shared module designation file exists, identifying at least one shared web module from the shared module designation file to be incorporated into a application to form at least one identified shared module (Col. 8, lines 10 – 21; Col. 8, lines 50-55; Col. 10, lines 26 - 31), wherein the shared designation file includes all descriptors that reference the at least one shared module (Col. 9, lines 16 – 32);

locating the at least one identified shared module using path information (Col. 8, line 67- Col. 9, line 4; Col. 9, lines 16-32); and

logically merging the at least one shared module with modules of the web

application, in accordance with the shared module designation file to generate a

logically merged web application (Col. 10, lines 26 – 49), wherein a reference to the at

least one shared web module is used in the logically merged application rather than a

copy of the at least one shared module (Col. 8, lines 22-26).

Li does not explicitly indicate that the applications and modules comprise web

applications and its modules.

Spotswood teaches that web applications can be run using java virtual machine

on a web server (¶56-62).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to use Spotswood's teaching of web applications and their use of

web modules and java classes to expand Li's teaching of shared java classes to include

web applications running in java virtual machine.

Regarding claims 2, 12, and 20, Li teaches the method of claims 1, 11, and 19,

further comprising: loading the logically merged application into a container (Col. 9, lines

5 – 15).

Li does not explicitly indicate a web application running on a web container.

Spotswood teaches that web applications can be run using java virtual machine

on a web server (¶56-62), where the applications are run using on a web container

(¶76).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to use Spotswood's teaching of web applications and their use of

web modules and java classes to expand Li's teaching of shared java classes to include

web applications running in java virtual machine.

Regarding claim 4, Li teaches the method of claim 1.

Li does not explicitly indicate wherein the web application is an enterprise archive

(EAR) and wherein the logically merged web application is a logically merged EAR.

Spotswood teaches that web applications can be run using java virtual machine

on a web server (¶56-62), indicate wherein the web application is an enterprise archive

(EAR) and wherein the logically merged web application is a logically merged EAR

(¶49-53).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to use Spotswood's teaching of web applications and their use of

web modules and java classes to expand Li's teaching of shared java classes to include

web applications running in java virtual machine.

Regarding claims 6 and 14, Li teaches the method of claims 1 and 11, wherein

logically merging the at least one shared web module with web modules of the web

application includes: determining a priority associated with the at least one shared web

module (Col. 9, lines 59 – 67); and resolving any conflicts between shared web modules

in the at least one shared web module and conflicts between the at least one shared

web module and web modules of the web application, if any (Col. 8, lines 10 – 21).

Regarding claims 7 and 15, Li teaches the method of claims 1 and 11, wherein

the steps of identifying, locating, and logically merging are performed during an

initialization process of a runtime environment for initializing the web application to be

run on a server (Col. 8, lines 40 – 62).

Regarding claim 8, Li teaches the method of claim 1, wherein logically merging

the at least one shared web module with the web modules of the web application

includes using a service provider interface (SPI) that provides merge logic for merging

different module types (Col. 8, lines 40 – 62).

Regarding claims 9 and 17, Li teaches the method of claims 2 and 12, wherein

the container uses one or more application program interfaces (APIs) to identify a path

to the at least one shared web module and loads the at least one shared web module

when loading the logically merged web application (Col. 9, lines 5 – 9).

Regarding claims 10 and 18, Li teaches the method of claims 1 and 11,

wherein logically merging the at least one shared web module with web modules of the

web application includes at least one of re-linking references to the at least one shared

web module in the web modules of the web application, extrapolating policy information

for the at least one shared web module from a policy file associated with the web

application, and modifying a class path for the web application to include paths to each

of the at least one shared web modules (Col. 9, lines 16 – 31; lines 61 – 67).

Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Li in

view of Spotswood, and further in view of Sharma (6721777).

Regarding claim 5, the combined teaching of Li and Spotswood disclose

method of claim 1.

The combination does not explicitly indicate wherein the at least one shared web module includes at least one of a web archive (WAR) file, an enterprise java bean (EJB) archive file, and a resource archive (RAR) file.

Sharma teaches that in addition to the java classes that are shared between applications taught in Li, that web applications can also share resource adaptor modules (Col. 3, lines 13 – 15; lines 29-39).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include these resource adapters in Li's shared memory space to ensure that memory usage gets saved by only running one resource adapter at a time and sharing it between applications.

(10) Response to Argument

Regarding claims 1, 11, and 19, the appellant argues that Li does not teach descriptors which reference at least one shared module. More particularly, the links presented in the name table of Li does not anticipate the reasonable interpretation of a descriptor. See Appellant's Brief pg 9-10.

The examiner disagrees:

The appellant argues that a descriptor is typically characterized as a file which describes the contents or the functionality of another object. *See* Appellant's Brief pg 9-10. Descriptors are solely described in the specification on pg 19, line 11 through pg 20, line 14. It is clear from that disclosure of the descriptors is that they function more of

a way to identify and locate shared web modules which operate for a web application.

See Spec. pg 20, lines 4 – 10. The specification provides no indication that a descriptor

is necessarily a file nor has anything to do with objects as the appellant has argued.

Li teaches a method of sharing java class among web applications operating on

the same container. *See* Li, Col. 7, lines 31 – 41. Li teaches that multiple JVMs which

each operate a plurality of classes and those classes can be identified through a name

table if they are currently operating in the shared memory pool. *See* Col. 9, lines 25 -

31. Where when the class is initialized, it is created with a name and gets a path within

the shared memory pool. *See* Li, Col. 10, lines 20 – 26. Finally, when the JVM needs

to operate a certain class, it provides a call for that class in order to identify the

existence of the classes within the shared memory. *See* Col. 10, lines 26 – 49. It is

clear from Li, that the classes which are to be shared have the ability to be identified

and found by the JVM which is attempting to run the class, and by any other JVM which

attempts to share that class. As result, each application must be equipped with the

proper descriptor or reference to the valid name of the class within the system and

shared memory pool.

Regarding claims 1, 11, and 19, the appellant argues that Li does not teach

logically merging at least one shared web module. More specifically, that logically

merging should be construed as merging objects in logic without creating a physically

merged object, while Li only teaches "sharing an instantiation".

The examiner disagrees:

The appellant's description of what logical merger consists of is in direct disagreement with the specification. The process of merging web modules to be shared is described in the specification on pg 13, line 21 though pg 14, line 29. The specification describes how applications and shared modules get merged is through the reuse of certain modules which two separate applications both use as part of their operation. Li teaches a system of operating a plurality of JVM applications and which have the ability to share common java classes among themselves. *See* Li, Col. 10, lines 26 - 49. As result, the sharing of classes in Li relates to the logical merger of classes which both JVMs both need to operate and allow them to both operate while only having one single copy of the needed java classes. *See id.* Thus Li teaches the limitation recited in the claim in light of the specification, rather than the dictionary definition the appellant is attempting to assert.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/KEVIN BATES/

Primary Examiner, Art Unit 2456

Conferees:

/Aaron Strange/
Primary Examiner, Art Unit 2448

/John Follansbee/
Supervisory Patent Examiner, Art Unit 2451